



Omnis Developer Conference



Mejora tu desarrollo:

Nuevas funciones en Omnis Studio 11.2
que elevan el nivel de tus aplicaciones

Francisco Ramos

JavaScript Remote Forms

Variable parámetro Row del Construct

La variable parámetro de tipo row del \$construct en una remote task, ahora contiene dos columnas adicionales, la primera denominada “**headers**”, se trata de una row donde cada columna contiene un par nombre-valor por cada ítem del encabezado HTTP recibido del cliente.

Además de ésta, el parámetro row muestra otra nueva columna adicional denominada “**cookies**”, se trata también de una row donde cada columna contiene un par nombre-valor por cada cookie, por ejemplo, cookie1=hello dará como resultado una row con 'cookie' como nombre de columna y 'hello' como valor.

Subformset Panels

Funcionamiento similar al subform control.

- \$panelcount
- \$currentpanel
- \$panelclassname
- \$closebutton
- \$esctoclose
- \$openmax
- \$panelsubformparams

Title 1	▼ ×
Title 2	^ ×
Subform text for Panel 2	
Title 3	▼ ×

Atajos de teclado personalizables

Ahora es posible interceptar y modificar la entrada del teclado en remote forms o controls utilizando un controlador de eventos personalizable.

\$keyboardshortcut (en una class o field)

pSystemKey, pCharacterKey, pModifiers, pFocusedControl

```
If pCharacterKey="s"&|ControlKey=kTrue
```

```
  Do $cinst.$save()
```

```
End If
```

Copia de datos al portapapeles

- `$copytoclipboard(vDataToCopy[,cMIMEType='text/plain',bDatalsBinary=kFalse])`

On evClick

Do `$cinst.$copytoclipboard(iText)`

On evClick

Do `$cinst.$copytoclipboard(iText,"text/html")`

On evClick

Do `$cinst.$copytoclipboard(ilImageData,"image/png",kTrue)`

Lectura de datos del portapapeles

\$readclipboard

Este método en un remote form nos permite leer el último contenido del portapapeles del usuario final, el cual a su vez invocará al método...

\$onclipboardread remote form

...pasándole tres parámetros : pSuccess, pPlainText, pExtraData

pExtraData es una row que contiene el tipo MIME asociado a los datos (por ejemplo 'text/html', 'image/png').

Métodos `_return`

Al invocar un método "server-execute" desde el cliente, éste normalmente devolverá una promesa (promise), a menos que exista un método "..._return" coincidente, en cuyo caso se invocará al método de retorno.

El valor de retorno para el método en el servidor se pasará como parámetro en la función "**then**" de la Promesa. Esto permite encadenar una lógica asíncrona en el mismo método, manteniendo el contexto y utilizando las variables definidas en el método externo. Por ejemplo:

```
Do $cinst.$myServerMethod() Returns IPromise
```

```
JavaScript: IPromise.then( returnVal) => {
```

```
JavaScript: IRetVal = returnVal; // Asigna el parámetro a una variable local que Omnis  
puede reconocer
```

```
Do $cinst.$showmessage(IRetVal,"Método de servidor completado")
```

```
JavaScript: });
```

Notificaciones Push - openpush

El client command 'openpush' (ejecutado mediante \$clientcommand) tiene dos nuevos parámetros row() opcionales iMaxTries y cRetryCodes.

- **iMaxTries**

Número máximo de intentos automáticos de una solicitud, en caso de que se devuelva un código de estado HTTP coincidente con cRetryCodes. -1 significa ilimitado. El valor predeterminado es 5.

- **cRetryCodes**

Códigos de estado HTTP considerados como "errores temporales". Se reintentará automáticamente hasta un máximo de iMaxTries, antes de reportar un error. Esta cadena contiene códigos separados por comas o rangos de códigos; por ejemplo, "404,500-599" significa que cualquier respuesta con el código de estado HTTP 404 o entre 500 y 599 se reintentará automáticamente. El valor predeterminado es 500-599.

Notificaciones Push - \$pushclosed

Un nuevo método “client-executed”, para los “remote forms” denominado \$pushclosed, es invocado en todos los formularios en el caso de que se cierre una conexión push.

Se deberá retornar kTrue desde el método para evitar el comportamiento predeterminado.

\$pushclosed tiene dos parámetros:

- **pErrorCode**

Valores distintos de cero indican que se cerró debido a un error.

0 significa que fue cerrado por invocación del comando \$clientcommand 'closepush'.

> 1 significa que pErrorCode es el código de estado HTTP que causó el error. Se mostrará un mensaje de error, excepto si retronamos kTrue evitando el comportamiento predeterminado.

-1 significa un error de transporte, es decir, no se pudo entregar la solicitud.

- **pWillRestart**

Si es verdadero, el comportamiento predeterminado será reiniciar la conexión push.

Se deberá retornar kTrue para evitarlo.

Subform Palettes - Palette Arrow

Al usar el “client command” "subformpaletteshow" para abrir un “subform palette”, podremos ahora utilizar el indicador **kSFSPaletteHideArrow** en el parámetro row **iPositionFlags** para ocultar la flecha que normalmente se muestra en la ventana de la paleta.

Subform Palettes - Control Name

Si no existe un nombre del control proporcionado en el parámetro cControl del "client command" "subformpaletteshow" (o bien es omitido), el cuadro de diálogo de la paleta se centrará sobre la ventana del navegador y la flecha no se mostrará, ya que no hay ningún control al que pueda apuntar.

Subform Palettes - Cut out

Se ha agregado un “remarque” alrededor del objeto para los cuadros de diálogo de los “palettes” cuando se muestra en superposición.

Si se pasa el parámetro **bShowOverlay** como *kTrue* en el “client command” “subformpaletteshow” (\$clientcommand), entonces y si se apunta a un control (o un sub-elemento del control), se mostrará un “remarque” alrededor del elemento para resaltarlo.

Date Parsing

El analizador de fechas de Omnis intenta convertir la representación textual de una fecha o fecha/hora al formato de fecha/hora de Omnis. Se ha mejorado el análisis de cadenas de fecha y ahora soporta formatos de fecha/hora cortos..

El formato corto de fecha y hora depende de la configuración regional del navegador; por ejemplo, para inglés británico es D/M/y, y para inglés estadounidense es M/D/y. Por lo tanto, un usuario final podría introducir "3/5" en cualquier campo de fecha o fecha/hora y se convertiría a "3 May 2024" o "March 5 2024".

Reescribir el historial del navegador

Dos nuevos métodos para formularios nos permiten ahora interactuar con el historial del navegador y agregar eventos de navegación activados mediante -clics- en los botones Atrás o Adelante del navegador.

\$pushhistorystate(wRow, [bReplaceState=kFalse])

\$applynewhistory

recibe la row “wRow” de \$pushhistorystate como parámetro.

Componentes JavaScript

JS Entry Field Border Icons

\$bordericonstyle

\$setbordericonstyle

evBorderIconClicked

Se pueden mostrar iconos al borde de los campos de entrada (también en *Combo boxes* y *Drop lists*) usando la propiedad **\$bordericonstyle**, sólo aplicable si el estilo del borde en \$effect es kJSborderPlain.

Name

Email

Phone

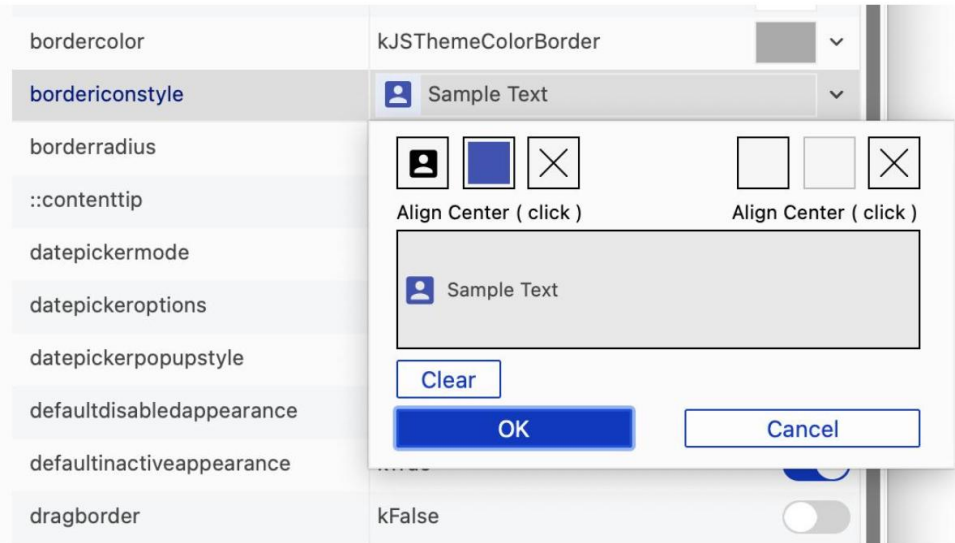
Address

JS Entry Field Border Icons

Una única propiedad guarda la configuración de íconos izquierdo y derecho, accesible desde el Property Manager.

Se puede usar el evento **evBorderIconClicked** para activar los iconos del borde. Este evento contiene el parámetro *pLeftBorderIcon*.

Si es *true*, se hizo clic en el icono del borde izquierdo; si es *false*, se hizo clic en el derecho.



JS Data Grid - \$filteronkeypress

Si **\$filteronkeypress** es *true*, la rejilla es filtrada con cada pulsación de tecla

(valor por defecto, para mantener su comportamiento habitual).

Si es *false*, el filtrado se realiza tras presionar la tecla Enter o bien cuando el foco abandona el campo.

JS Data Grid - Grid Cell Tooltips

Ahora es posible mostrar información sobre celdas individuales en un Data Grid.

El método **\$gettooltip()**, (client-executed), es invocado al pasar el cursor sobre una celda. El texto devuelto se muestra como información sobre la celda, sus parámetros *pHorzCell*, *pVertCell* y *pDataColumnName* permiten identificar la celda sobre la que se ha pasado el cursor.

JS Data Grid - Drop events

Los eventos *evDrop*, *evCanDrop* y *evWillDrop* contienen un nuevo parámetro *pColNum* el cual informa del número de columna donde se soltaron los datos.

El control “Data Grid” incluye éste nuevo parámetro, *pColNum*, disponible para los controladores de eventos indicados.

JS Data Grid - Pick Lists

Se ha agregado la propiedad **\$picklistrowheight** al control “Data Grid” para permitir fijar la altura de la fila en las “pick lists” (menú de selección).

Podrá asignarse el mismo valor de altura que ya tiene la fila en el “Data Grid”, por ejemplo, 44, que es la altura predeterminada.

La altura de la fila en la “pick list” no podrá ser menor que la altura del tipo de letra en uso.

Datagrid - \$enablecell()

Se ha añadido el método **\$enablecell** al control “Data Grid” para evitar que celdas específicas puedan ser editadas.

El método `$enablecell(pRow,pCol,pDataColumnName)` es invocado antes de que una celda se haga editable, por lo que devuelve `kFalse` al deshabilitar la edición de la celda.

Este método puede ser ejecutado tanto desde el cliente, como desde el servidor.

JS Data Grid - Date Picker

Si alguna de las propiedades del "Data Grid" `$editdatetext` o `$columneditdatetext` es *true*, el botón "Date Picker" ya no se considera como una tabulación independiente. Por tanto, podrá abrirse el "Date Picker" pulsando "*Alt + Flecha abajo*" o haciendo clic sobre él.

Se puede mantener el comportamiento anterior, mediante fijar la variable JavaScript '*alwaysTabToDatePicker*' como *true* para el "Data Grid". Por ejemplo:

```
Calculate IDataGrid as $cinst.$objs.datagrid  
JavaScript:IDataGrid.alwaysTabToDatePicker = true;
```

Native List - \$extrascrollheight

La propiedad **\$extrascrollheight** especifica la altura del espacio adicional (cuando se hace scroll) que se agrega al final de una "Native List".

Por ejemplo, esto puede ser de utilidad a fin de dejar espacio para mostrar un botón de acción flotante (Action Button) sobre la parte superior de una "Native List".

Native List - Buttons

Ahora es posible agregar botones en los encabezados de grupo de "Native List", además se puede indicar una URL de ícono (o imagen) como su contenido.

Fijar **\$groupaccessorytypecol** como *kJSNativeListAccessoryTypeButton* permite agregar un botón al encabezado del grupo (actualmente no se admite ningún otro tipo de accesorio).

\$groupaccessorycontentcol se puede configurar como texto o como una URL de icono/imagen; la imagen se detecta porque la URL termina con una extensión .png/.jpg/.jpeg/.svg.

Screens	Navigation
Tab Navigation	+
  Home home	
  Users admin	
App Menu	+
  View Stock Stock	

Native List - Buttons

Cuando se haga -clic- sobre un accesorio del encabezado de grupo, se activará *evClick*.

El parámetro **pWhat** contendrá **kJSNativeListGroupAccessory** y **pGroup** será el número del grupo.

Scroll Shadows for lists

Las sombras de los bordes ahora resaltan el contenido desplazable en *lists*, *grids*, *containers* y otros controles.

\$scrollshadowwidth

\$scrollshadowcolor

wList

wTest

wTest2

wTestFTP

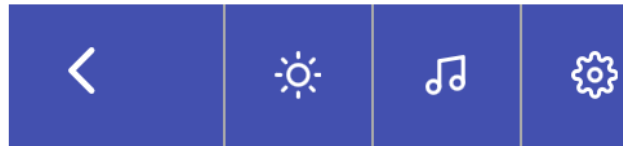
Toolbar Control back button

Ahora se podrá añadir un botón "Atrás" a la izquierda de un control "JS Toolbar" mediante fijar su propiedad `$backbutton` en `kTrue`. En este caso y si el menú lateral está visible (`$sidemenu` es `kTrue`), el menú mostrará el botón Atrás. Cuando se hace -clic- sobre él, se activa un nuevo evento denominado `evBackClick`.

`$backbutton` = `kTrue`

`$event`

On `evBackClick`



JS Tab Bar Control - \$currenttab

La propiedad **\$currenttab** de un control "Tab Bar" ahora puede ser fijado a 0.

Esto tiene el efecto de deseleccionar todas las pestañas.

Html Link Control - Link URL

Si las propiedades *\$linkurl*, *\$linkurlname* o *\$linkedobject* están **vacías**, la URL para el control “HTML Link” toma como valor predeterminado el texto de *\$text* o *\$dataname*, siempre que el contenido tenga un formato de URL válido (<http://...> o <https://...>).

Eventos

El comportamiento predeterminado para *evClick* del control “HTML Link” ahora se puede ser anulado mediante el comando “Quit event handler (Discard event)”.

Date Pickers

Las ventanas emergentes del selector de fecha ahora usan la configuración regional predeterminada del cliente para el primer día de la semana.

Los selectores de fecha para los “Entry fields” y “Data Grids” (excepto para el control “Date picker”) ahora usan el primer día de la semana estándar de la configuración regional del navegador del cliente.

Anteriormente, el primer día de la semana era el domingo.

Navigation Menu Object

Se han añadido dos nuevas propiedades, **\$::iconcolor** y **\$cascadeiconcolor**, al control “Navigation Menu” para permitir configurar el color de los iconos SVG. Esta mejora se aplica tanto los controles “Navigation Menu” JavaScript (Remote form) como a los “Nav Menu” en las clases “Window”.

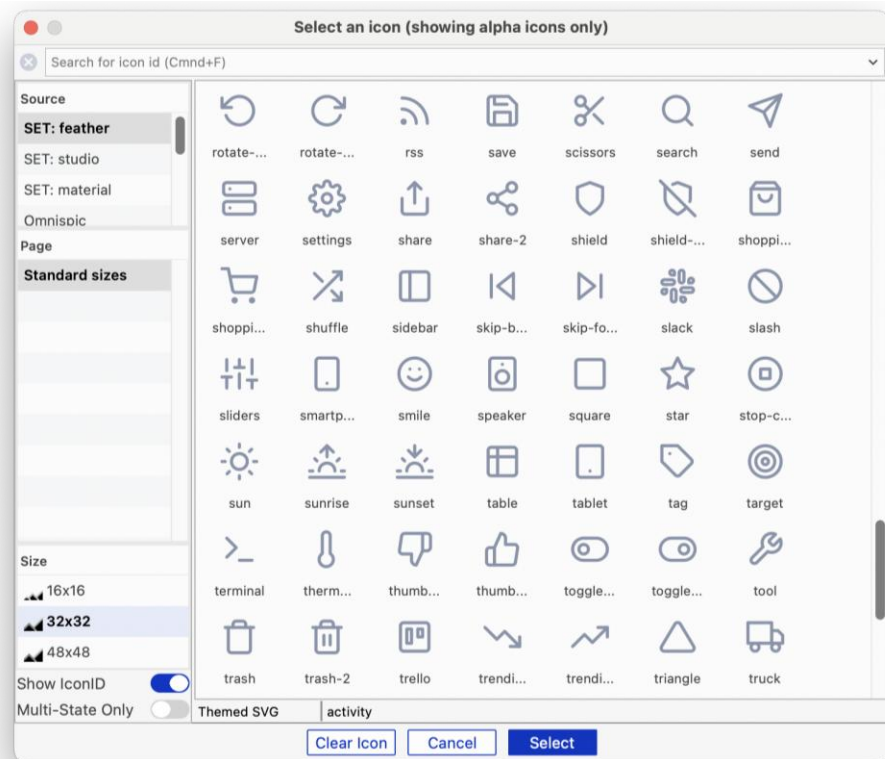
La propiedad **\$::iconcolor** indica el color del SVG en el menú de nivel superior especificado en **\$verticalcascadeiconid**.

La propiedad **\$cascadeiconcolor** especifica el color de los SVG en los menús en cascada, especificados en **\$horizontalcascadeiconid**, **\$closeboxiconid** y **\$shotcloseboxiconid**.

Para componentes JavaScript, *kColorDefault* ajusta las reglas de color para texto predeterminadas, mientras que en clases “Window”, *kColorDefault* causa que los colores de los íconos sigan el color del texto.

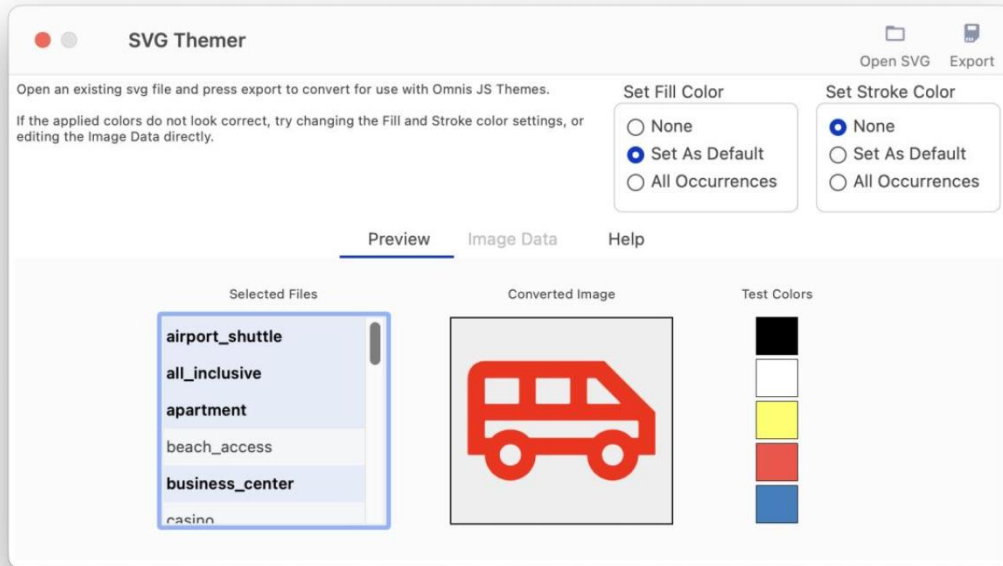
Nuevo conjunto de iconos "Feather" (pluma)

Agregue "feather" a su librería mediante su propiedad **\$iconset**.



SVG Themer (Herramienta para el manejo de temas SVG)

Mediante “SVG Themer” puede crear temas exportando el SVG convertido para su uso con Omnis JS.



Implementación de aplicaciones

Server Configuration - discardRequestsContaining

Se ha añadido el elemento **discardRequestsContaining** a la sección "server" en el archivo de configuración de Omnis.

DiscardRequestsContaining es una matriz de textos. El servidor Omnis comparará cada texto con la URL solicitada. Si coincide, la solicitud es descartada. Por defecto, la matriz contiene lo siguiente:

```
"discardRequestsContaining": [  
  "../"  
]
```

Esto significa que cualquier solicitud recibida por el servidor Omnis que contenga "../" en cualquier parte de la URL será descartada y se devolverá un error al cliente.

Firstruninstall

Se ha agregado una función oculta para evitar que la carpeta "firstruninstall" sea copiada a la carpeta de datos si ya existe (Windows o macOS).

Omnis ahora busca un archivo denominado "**disable_fri.txt**" en la raíz de la carpeta de datos. Si existe, no se copia la carpeta "firstruninstall". No se lee el contenido del archivo, solo se comprueba su existencia.

Componentes para clases “Window”

Entry Fields y Combo Boxes

Se han añadido las propiedades **\$labelcolor**, **\$labelfontsize** y **\$labelhottextcolor** a los “Entry fields” y “Combo boxes” para controlar el color y el tamaño del texto de las etiquetas de sugerencias de contenido flotantes.

De manera predeterminada, contienen un tamaño para la fuente de “0” y “*kColorDefault*” para el color, de éste modo se preserva su comportamiento anterior, donde el color del texto de la etiqueta es el mismo que el del campo y el tamaño de la fuente es el 80% de la del campo.

Masked Entry Field - \$passwordchar

Se ha añadido la propiedad **\$passwordchar** a los “Masked Entry Field”.

Puede usar '*' en el \$passwordchar de un “Masked Entry Field” para reemplazar los caracteres por asteriscos y así ocultar el contenido del campo.

Pushbuttons

El estilo de los botones configurados como *kSystemButton* en *\$buttonstyle* ha cambiado para macOS.

Para las “headed lists” y “tree lists”, *\$buttonstyle* es el estilo del “header button”, ya sea *kSystemButton* o *kUserButton*. El estilo del botón *kSystemButton* ahora usa el estilo de un botón pulsador nativo del sistema con una altura flexible. Aún si la altura supera la de un botón de tamaño normal se mantiene el mismo estilo.

En versiones anteriores, se usaba el estilo de sistema cuadrado cuando la altura era mayor que la de un botón de tamaño normal. Para volver a este comportamiento anterior, establezca la nueva preferencia raíz *\$squaresystembuttons* en *kTrue*..

Headed List - `$showcolumnlinesifempty`

Se ha agregado una nueva propiedad **`$showcolumnlinesifempty`** a las “Headed Lists” para mostrar las líneas de columna aún cuando la lista está vacía.

Esto sólo es aplicable cuando `$showcolumnlines` es *kTrue*.

oBrowser - \$allowscrollzoom

La propiedad `$allowscrollzoom` permite alternar la funcionalidad de zoom con desplazamiento dentro de una ventana del navegador OBrowser.

Cuando es `kTrue` (valor predeterminado), el usuario final puede acercar o alejar el contenido del navegador manteniendo presionada la tecla “Ctrl” mientras mueve la rueda del mouse.

Si `$allowscrollzoom` es `kFalse`, dicha función estará deshabilitada.

Border Icons

Los “border icons” ahora responden a los clics. Aplicable sólo a los controles compatibles con `$bordericonstyle`, tales como los “Edit fields”, “Token edit fields”, “Combo boxes” y “Drop lists”.

On evBorderIconClicked

 If pLeftBorderIcon

 Popup menu MyOptionsMenu

 Else

 Send to trace log Se ha pulsado sobre el icono derecho

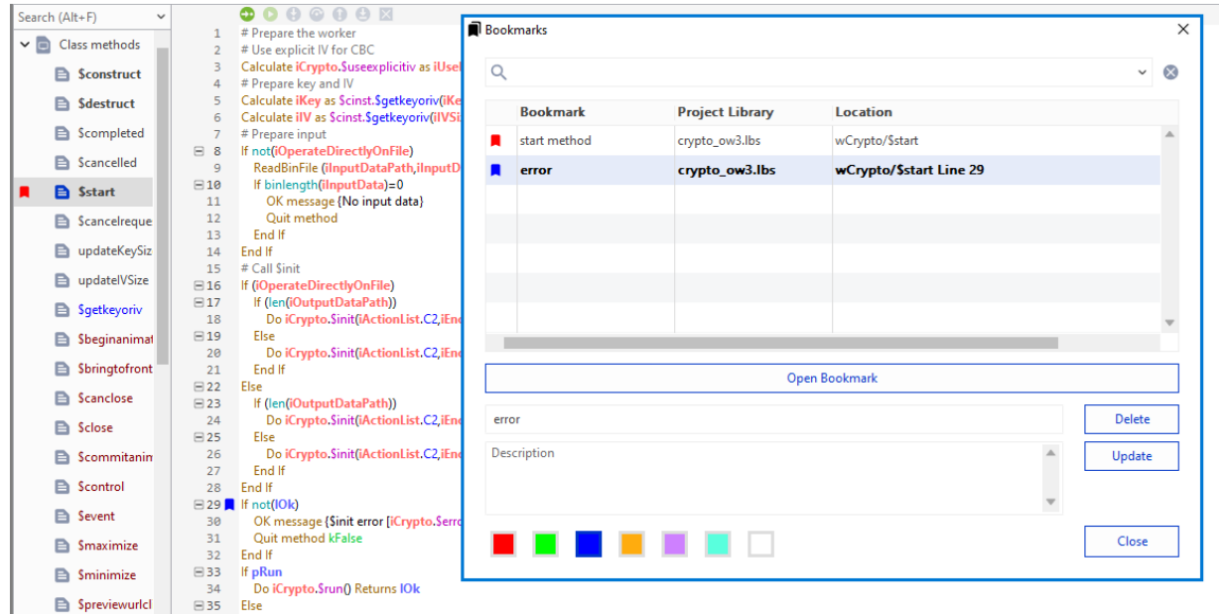
End If

The background is a solid blue color with several diagonal stripes of varying shades of blue running from the top-left to the bottom-right. A single, thin, light blue line is also visible, running diagonally across the upper left portion of the image.

Editor de métodos

Marcadores de método (bookmarks)

Ponga marcadores en métodos o líneas para marcar y de éste modo encontrar más fácilmente código importante.



Librerías y Clases

Exportando librerías a JSON

Las siguientes propiedades relacionadas con VCS se excluyen de la exportación a JSON cuando la opción *fullexportimport* (parte de la preferencia raíz *\$exportimportjsonoptions*) es “false”:

- Propiedades de las librerías:

\$vcsbuilddate, *\$vcsbuildersname*, *\$vcsbuildnotes*

- Propiedades de las clases:

\$vcsrevision, *\$showascheckedout*

File Classes

El editor para “File Class” ahora incluye el “userinfo” de cada campo en la clase, puede usarse para guardar datos de cualquier tipo relacionados con él.

Es asignado como `fileclass.field.$userinfo`.

Description	my primary key
UserInfo	use this for any additional information

Informes (reports)

Campos editables en informes

Marcar la propiedad **\$enterable** de un campo a *kTrue* permite que los usuarios ingresen y guarden texto en un informe PDF.

Sólo con archivos PDF generados mediante “Omnis PDF Device”.

\$setdocinfo

El método **\$setdocinfo** ahora permite indicar el productor y creador del archivo PDF de salida.

Sólo con archivos PDF generados mediante "Omnis PDF Device".

```
$setdocinfo(cAuthor[,cTitle="",cSubject="",cKeywords="",cLanguage='en-GB',cProducer='Omnis Studio PDF Device',cCreator='Omnis Studio'])
```

cProducer

Es la aplicación que genero el PDF, por defecto "Omnis Studio PDF Device".

cCreator

Es la aplicación que generó el PDF, por defecto "Omnis Studio".

\$embedfile() y \$embeddata()

Se han añadido dos nuevas funciones a “Omnis PDF Device” las cuales permiten adjuntar archivos y datos en documentos PDF.

Sólo con archivos PDF generados mediante “Omnis PDF Device”.



Omni VCS API

VCS API - \$getapiobject()

La forma de acceder a la API de VCS ha cambiado. Ahora se deberá usar el método **\$getapiobject** para acceder a sus métodos.

Para invocar métodos de la API, primero deberá usarse **\$getapiobject()** el cual devolverá un “object reference” a la API de VCS, por ejemplo:

Do \$root.\$modes.\$getapiobject('VCS') Returns iAPIObjRef

Una vez obtenido el “object reference”, todas las llamadas a la API podrán realizarse utilizando dicha referencia, lo que simplifica su sintaxis. Por ejemplo, para invocar al método \$login, podrá usarse:

Do iAPIObjRef.\$login(cUSERNAME, cPASSWORD, nTokenTime, cToken, cErrors, rSessionOrSessionPoolRef, cDBName) Returns bStatus

VCS API - \$buildProject

El método API VCS *\$buildProject* ahora permite bloquear o excluir clases de la compilación.

El método *\$buildProject* contiene dos nuevos parámetros de tipo “list”, **excludeClassList** y **lockClassesList**.

Ambos se definen mediante **cClassName** y **cClassType** (por ejemplo, wTest, kWindow) y deberá contener una lista de las clases que se excluirán o de las que se bloquearán.

OW3 Worker Objects

LDAP Worker Object

El método *\$init* del worker LDAP ahora permite los parámetros *cUser* y *cPassword* de tipo carácter o binario; esta mejora es establecida para permitir nombres con caracteres especiales.

Cuando se use "Character", Omnis enviará *cUser* y *cPassword* en formato UTF-8. Cuando se use "Binary", se enviarán con la codificación que requiera el servidor; por ejemplo, algunos servidores requieren CP1252.

HTTP Worker - Uso de IA en Omnis

Es posible acceder a los modelos de IA a través de su API mediante Omnis HTTP Worker.

Se ha agregado una aplicación de ejemplo denominada “IA” a la sección “Samples” del Hub en el navegador de Studio para demostrar el acceso a varios modelos de IA.

HASH Worker - \$initsignature

El método *\$initsignature* contiene un nuevo parámetro opcional *iSignatureType* el cual permite seleccionar el tipo de firma, RSA o ECDSA.

Además de esto, el parámetro *bBlind* está configurado para que ahora y de forma predeterminada sea *kTrue*.

OAUTH2 Worker - \$pkceentropy

Se ha añadido la propiedad **\$pkceentropy** al worker OAUTH2 para controlar la longitud del verificador de código para PKCE.

La propiedad **\$pkceentropy** permite un valor entero entre 43 (por defecto) y 128.

Funciones

Searching Lists - bSetCurLineWhenNotSelecting

```
$search(calculation [,bFromStart=kTrue, bOnlySelected=kFalse,  
bSelectMatches=kTrue, bDeselectNonMatches=kTrue,  
bSetCurLineWhenNotSelecting=kTrue])
```

Cuando **bSetCurLineWhenNotSelecting** es **false** (su valor por defecto es *kTrue*) y las opciones de selección/de-selección son también false, la búsqueda se completa devolviendo el número de línea encontrada, pero sin ser seleccionada.

FileOps

\$copy() y **\$move()**

- funciona con archivos y carpetas.
- Incluye nuevos parámetros para crear rutas faltantes, reemplazar archivos existentes y fusionar contenido.

\$joinpath(cSegment, ...)

- Une todos los segmentos indicados utilizando el separador específico de la plataforma en uso como delimitador.

FileOps.\$writefile - bCreateParentDirectories

Se ha añadido un nuevo parámetro **bCreateParentDirectories** a la función *FileOps.\$writefile* para permitir la creación de los directorios especificados en la ruta.

```
FileOps.$writefile(cFilePath,vVariable [,iEnc=kUniTypeUTF8, bBOM=kTrue,  
bReplace=kTrue, bCreateParentDirectories=kFalse])
```

bool()

Nueva función **bool()** la cual permite convertir un valor a booleano.

Do bool(#NULL) ## devolverá kFalse

If kTrue | bool(#NULL) ## devolverá kTrue

...

End If

If bool(lObjRef) ## será kTrue si lObjRef apunta a una instancia o si no kFalse.

...

End If

idletime()

Nueva función **idletime()** la cual retorna el tiempo transcurrido en milisegundos desde la última vez que se utilizó el teclado o el ratón.

coalesce() y coalesceempty()

coalesce()

- Retorna el primer parámetro non-NULL de entre una serie.

\$coalesceempty()

- Retorna el primer parámetro non-NULL y non-Empty de entre una serie.

OREGEX

OREGEX.\$replace(cSource, cTarget, cReplace [, iFlags, &cErrorText]))

- Busca una coincidencia en cSource usando el patrón en cTarget y reemplaza la sub-cadena coincidente con cReplace.

OREGEX.\$replaceall(cSource, cTarget, cReplace [, iFlags, &cErrorText]))

- Busca todas la coincidencias en cSource usando el patrón en cTarget y reemplaza la sub-cadena coincidente con cReplace.

Los “flags” (banderas) pueden encadenarse, p. ej.
kORegexCaseInsensitive+kORegexMatchNotNull.

Métodos JSON Object

Algunos métodos de “JSON Object” contienen un nuevo parámetro para garantizar que los objetos faltantes sean creados automáticamente.

`$setstring()`, `$setbool()`, `$setinteger()`, `$setfloat()`, y `$setobject()`.

La sintaxis para `$setstring()` ahora es:

- `$setstring(cMember,cString[,bAddMissingMembers=kFalse])`

Fija el miembro especificado como cadena JSON, con el valor `cString`.

Cuando `bAddMissingMembers` se pasa como `kTrue`, se crean los miembros faltantes (solo objetos); el valor predeterminado es `kFalse`.

Retorna `"true"` si la operación se ha realizado correctamente.

rcedit

Los nuevos métodos estáticos para rcredit (solo Windows) los cuales devuelven diversa información sobre los ficheros.

- **\$getapplicationmanifest**(cFileName)
Retorna el "manifest" del exe/dll indicado en cFileName.
- **\$getfileversion**(cFileName)
Retorna la versión del exe/dll en cFileName.
- **\$getproductversion**(cFileName)
Retorna el "product version" del exe/dll en cFileName.
- **\$getresourcestring**(cFileName, iKey)
Retorna el "resource string" con clave iKey del exe/dll en cFileName.
- **\$getversionstring**(cFileName, cKey)
Retorna el "version string" con clave cKey del exe/dll en cFileName.

**Y... muchas más
mejoras menores**